Jhon Steve Gómez

Diseño electrónico

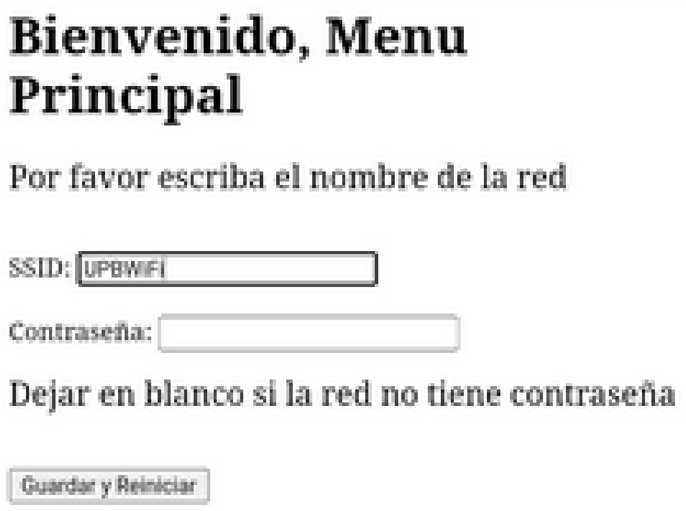
2024-1

Practica 4

Código

```
#include <Arduino.h>
#include <WiFi.h>
#include <WebServer.h>
#include <stdlib.h>
#include "TFT_eSPI.h"
#include <User_Setups/Setup25_TTGO_T_Display.h>
#include "data.h"
#include "Settings.h"
#include "UbidotsEsp32Mqtt.h"

#define DHTPIN 27
#define DHTTYPE DHT11
#define BUTTON_LEFT 0        // btn activo en bajo
#define LONG_PRESS_TIME 3000 // 3000 milis = 3s
#define MI_ABS(x) ((x) < 0 ? -(x) : (x))
WebServer server(80);

Settings settings;
int lastState = LOW; // para el btn
int currentState;     // the current reading from the input pin
unsigned long pressedTime = 0;
unsigned long releasedTime = 0;
const char *UBIDOTS_TOKEN = "BBUS-dLEMfK2bLu4HTl71h9LvpvL7NiVfu2"; // Put here
your Ubidots TOKEN
const char *DEVICE_LABEL = "esp32";                                // Put here
your Device label to which data  will be published
const char *VARIABLE_LABEL1 = "sw1";
const char *VARIABLE_LABEL2 = "sw2";
const char *TEMPERATURA_VARIABLE_LABEL = "Tempe"; // Temperatura
const char *HUMEDAD_VARIABLE_LABEL = "Hume";      // humedad

const int PUBLISH_FREQUENCY = 10000; // Update rate in milliseconds
unsigned long timer;

Ubidots ubidots(UBIDOTS_TOKEN);

void load404();
void loadIndex();
void loadFunctionsJS();
void restartESP();
void saveSettings();
bool is_STA_mode();
void AP_mode_onRst();
void STA_mode_onRst();
void detect_long_press();

// Rutina para iniciar en modo AP (Access Point) "Servidor"
void startAP()
{
  WiFi.disconnect();
  delay(19);
  Serial.println("Starting WiFi Access Point (AP)");
  WiFi.softAP("wifi-Jhon", "jhon1234");
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);
}

void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}
// Rutina para iniciar en modo STA (Station) "Cliente"
void start_STA_client()
{
  WiFi.softAPdisconnect(true);
  WiFi.disconnect();
  delay(100);
  Serial.println("Starting WiFi Station Mode");
  WiFi.begin((const char *)settings.ssid.c_str(), (const char
*)settings.password.c_str());
  WiFi.mode(WIFI_STA);

  int cnt = 0;
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    // Serial.print(".");
    if (cnt == 100) // Si después de 100 intentos no se conecta, vuelve a modo
AP
      AP_mode_onRst();
    cnt++;
    Serial.println("attempt # " + (String)cnt);
  }

  WiFi.setAutoReconnect(true);
  Serial.println(F("WiFi connected"));
  Serial.println(F("IP address: "));
  Serial.println(WiFi.localIP());
  pressedTime = millis();
  // Rutinas de Ubidots
  ubidots.connectToWifi((const char *)settings.ssid.c_str(), (const char
*)settings.password.c_str());
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();
}

void setup()
{

  Serial.begin(115200);
  delay(2000);

  EEPROM.begin(4096);                 // Se inicializa la EEPROM con su tamaño
max 4KB
  pinMode(BUTTON_LEFT, INPUT_PULLUP); // btn activo en bajo

  // settings.reset();
  settings.load(); // se carga SSID y PWD guardados en EEPROM
  settings.info(); // ... y se visualizan

  Serial.println("");
  Serial.println("starting...");

  if (is_STA_mode())
  {
    start_STA_client();
  }
  else // Modo Access Point & WebServer
  {
    startAP();

    /* ========== Modo Web Server ========== */

    /* HTML sites */
    server.onNotFound(load404);

    server.on("/", loadIndex);
    server.on("/index.html", loadIndex);
    server.on("/functions.js", loadFunctionsJS);

    /* JSON */
    server.on("/settingsSave.json", saveSettings);
    server.on("/restartESP.json", restartESP);

    server.begin();
    Serial.println("HTTP server started");
  }
}

void loop()
{
  if (is_STA_mode()) // Rutina para modo Station (cliente Ubidots)
  {
    if (!ubidots.connected())
    {
      ubidots.reconnect();
    }
    if (MI_ABS(millis() - timer) > PUBLISH_FREQUENCY) // triggers the routine
every 5 seconds
    {
      ubidots.add(TEMPERATURA_VARIABLE_LABEL, "Tempe"); // Insert your variable
Labels and the value to be sent
      ubidots.add(HUMEDAD_VARIABLE_LABEL, "Hume");      // Insert your variable
Labels and the value to be sent
      ubidots.publish(DEVICE_LABEL);
      timer = millis();
    }
    ubidots.loop();
  }
  else // rutina para AP + WebServer
    server.handleClient();

  delay(10);
  detect_long_press();
}

// funciones para responder al cliente desde el webserver:
// load404(), loadIndex(), loadFunctionsJS(), restartESP(), saveSettings()

void load404()
{
  server.send(200, "text/html", data_get404());
}

void loadIndex()
{
  server.send(200, "text/html", data_getIndexHTML());
}

void loadFunctionsJS()
{
  server.send(200, "text/javascript", data_getFunctionsJS());
}

void restartESP()
{
  server.send(200, "text/json", "true");
  ESP.restart();
}

void saveSettings()
{
  if (server.hasArg("ssid"))
    settings.ssid = server.arg("ssid");
  if (server.hasArg("password"))
    settings.password = server.arg("password");

  settings.save();
  server.send(200, "text/json", "true");
  STA_mode_onRst();
}

// Rutina para verificar si ya se guardó SSID y PWD del cliente
// is_STA_mode retorna true si ya se guardaron
bool is_STA_mode()
{
  if (EEPROM.read(flagAdr))
    return true;
  else
    return false;
}

void AP_mode_onRst()
{
  EEPROM.write(flagAdr, 0);
  EEPROM.commit();
  delay(100);
  ESP.restart();
}

void STA_mode_onRst()
{
  EEPROM.write(flagAdr, 1);
  EEPROM.commit();
  delay(100);
  ESP.restart();
}

void detect_long_press()
{
  // read the state of the switch/button:
  currentState = digitalRead(BUTTON_LEFT);

  if (lastState == HIGH && currentState == LOW) // button is pressed
    pressedTime = millis();
  else if (lastState == LOW && currentState == HIGH)
  { // button is released
    releasedTime = millis();

    // Serial.println("releasedtime" + (String)releasedTime);
    // Serial.println("pressedtime" + (String)pressedTime);
    long pressDuration = releasedTime - pressedTime;

    if (pressDuration > LONG_PRESS_TIME)
    {
      Serial.println("(Hard reset) returning to AP mode");
      delay(500);
      AP_mode_onRst();
    }
  }
  // save the the last state
  lastState = currentState;
}
```

Implementación página web



Bienvenido, Menu Principal

Por favor escriba el nombre de la red

SSID: UPBWiFi

Contraseña:

Dejar en blanco si la red no tiene contraseña